DEVELOPMENT OF AN ANOMALY TREATMENT TOOL FOR HUMAN DEEP SPACE EXPLORATION MISSIONS

by Oscar Balcells Quintana

April 2020

Bachelor's Final Thesis

Submitted to the faculties of

Facultat de Matemàtiques i Estadística (FME) and
Escola Superior d'Enginyeria Industrial, Aeroespacial i Audiovisual de Terrassa (ESEIAAT)
of Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

In partial fulfillment of the requirements for the

Bachelor's Degree in Aerospace Engineering
Bachelor's Degree in Mathematics

Under the guidance of
Daniel Selva Valero, Texas A&M University

Para mi familia y amigos

Per aspera ad astra.

Abstract

Time sensitive anomalies will become a crucial obstacle towards mission success in future human deep space exploration enterprises -namely to the Moon, Mars and beyond- due to their inherent and prohibitive communication delays between spacecraft and mission control ground support.

This thesis describes the on-going development of a Virtual Assistant (VA) tool that aims to help the astronaut crew in the context of on board and real-time anomaly treatment, therefore enhancing their level of autonomy. Particularly, the tool focuses in resolving anomalies that are specific to the Environment Control and Life Support System (ECLSS) of a crewed spacecraft. In order to tackle the anomaly treatment process, it is divided in three distinguished but related stages: anomaly detection, anomaly resolution and anomaly response.

Several validation experiments for the said VA are also presented, including a plan to test it under high fidelity conditions at NASA's Human Research Exploration Analog (HERA) during campaign number six. For such experiments, a trained crew consisting of four members will be kept in a confined and isolated environment that aims to emulate the interior set up of a deep space exploration spacecraft.

Keywords: Virtual Assistant, Anomaly, Anomaly Treatment, Anomaly Detection, Anomaly Diagnosis, Autonomy, Deep Space Exploration, Manned Missions, ECLSS, HERA, NASA

AMS Classification Code: 0000

Resumen

En futuras empresas de exploración del espacio profundo -a la Luna, Marte o incluso más allá- las anomalías que requieran una respuesta a muy corto plazo serán un gran obstáculo para el éxito de la misión, debido principalmente a los largos retrasos que dichas empresas entrañan en las comunicaciones entre la nave espacial y control de misión con base en tierra.

Esta tesis describe el desarrollo (actualmente en curso) de un asistente virtual para ayudar a las tripulaciones de astronautas en el contexto del tratamiento de anomalías en tiempo real a bordo de la nave, aumentando así su nivel de autonomía. En particular, esta herramienta se focaliza en resolver anomalías relativas exclusivamente al Sistema de Control de Entorno y Asistencia a la Vida. Para encarar el problema del tratamiento de anomalías, tal proceso se ha dividido en tres etapas distinguidas: detección de la anomalía, diagnóstico de la anomalía y resolución de la anomalía.

Varios experimentos para validar el susodicho asistente virtual son presentados en esta tesis también, incluyendo un plan para testearlo en condiciones de alta fidelidad en el Análogo para la Investigación de la Exploración Humana de la NASA durante su sexta campaña. Para tales experimentos, una tripulación entrenada constituida por cuatro miembros será sometida a un entorno de confinamiento y aislamiento que trata de emular la distribución del interior de una nave para la exploración del espacio profundo.

Palabras clave: Asistente Virtual, Anomalía, Tratamiento de Anomalías, Detección de Anomalías, Diagnóstico de Anomalías, Exploración del Espacio Profundo, Misiones Tripuladas, NASA

Resum

A les futures empreses d'exploració de l'espai profund - a la Lluna, Mart o fins i tot més enllà- les anomalies que requereixin una resposta a molt curt termini seran un gran obstacle per a l'èxit de la missió, degut principalment als llargs endarreriments que aquestes empreses comporten en les comunicacions entre la nau espacial i control de missió amb base a terra.

Aquesta tesi descriu el desenvolupament (actualment en curs) d'un assistent virtual per a ajudar a les tripulacions d'astronautes en el context del tractament d'anomalies en temps real a bord de la nau, augmentant així el seu nivell d'autonomia. En particular, aquesta eina es focalitza en resoldre anomalies relatives exclusivament al Sistema de Control d'Entorn i Assistència a la Vida. Per a encarar el problema del tractament d'anomalies, tal procés s'ha dividit en tres etapes distingides: detecció de l'anomalia, diagnòstic de l'anomalia i resolució de l'anomalia.

Diversos experiments per a validar l'esmentat assistent virtual són presentats en aquesta tesi també, incloent-hi un pla per a testejar-lo en condicions d'alta fidelitat a l'Anàleg per a la Investigació de l'Exploració Humana de la NASA durant la seva sisena campanya. Per a tals experiments, una tripulació entrenada constituïda per quatre membres serà sotmesa a un entorn de confinament i aïllament que pretén emular la distribució de l'interior d'una nau per a l'exploració de l'espai profund.

Paraules clau: Assistent Virtual, Anomalia, Tractament d'Anomalies, Detecció d'Anomalies, Diagnòstic d'Anomalies, Exploracio de l'Espai Profund, Missions Tripulades, NASA

Acknowledgements

First of all, and most important, thanks to my family and friends, for their love and support.

Thanks to Dr. Daniel Selva, not only for his guidance during the thesis, but also for hosting me in his research group.

Thanks to the SEAK Lab people, who treated me so kindly and made the stay at College Station far more pleasant.

Thanks to all the people who participated in the HCAAM project. In particular, Antoni Virós deserves a special mention, since his teachings and patience have strongly contributed to this thesis.

Thanks to CFIS, for bringing me the opportunity to study two bachelor degrees and to carry my final thesis at a major STEM university.

Last but not least, thanks to the diverse sources of funding that have supported my education. These include Fundació CELLEX, which granted me one of my two bachelor degrees, Fundació Caixa Catalunya, which awarded my baccalaureate performance, and the MOBINT and CFIS mobility scolarships, which funded this research stage at the USA. Thanks also to the Texas A&M University and its staff, for hosting me during the development of this thesis.

Contents

List of Figures

List of Tables

Preface

The purpose of this document is to explain the author's contribution to the development of an anomaly treatment tool (referred as Daphne-AT) for human deep space exploration missions. It is important to bear in mind though that Daphne-AT is the result of a collaborative project, and that it was built upon a previously existing tool known as Daphne.

To provide a bit of context, a good approach is perhaps to introduce the following: first, the research group where Daphne-AT has been designed; and second, the said "parent" tool or "starting point".

The SEAK Laboratory

Daphne-AT has been entirely developed at the Systems Engineering, Architecture and Knowledge (SEAK) Laboratory at Texas A&M University, and consequently, within its research framework.

The SEAK Laboratory is a part of the Texas A&M Aerospace Engineering Department. The research carried at this lab merges the topics of space, Artificial Intelligence (AI) and design. More specifically, it focuses on the development of advanced tools to support the systems engineering process, with emphasis on early system design and system architecture.

Maybe the best way to summarize the lab's activity is by presenting its mission statement as it is on its webpage `https://www.selva-research.com/` :

> "To support the development of the next generation of complex space systems while studying and improving the system architecture and design processes."

In particular, regarding on how the object of this thesis relates to this framework, Daphne-AT could be described as a space system tool which relies on human-machine interaction, visual and data analytics, machine learning techniques and knowledge representation and reasoning in order to enhance deep space exploration human missions feasibility.

Daphne

Daphne is a Virtual Assistant (VA) for aerospace system engineers, and its purpose is to help designers on the first stages of architecting Earth Observing Satellite Systems (EOSS) by acting as a peer in the design process and reducing cognitive workload. Its implementation is open-source, and it is available at `https://github.com/seakers` .

The first version of Daphne was a result of Antoni Virós' Bachelor's Final Thesis [1], titled *Design and development of a cognitive assistant for the architecting of earth observing satellites*. Although such document is currently outdated, it is a good resource to get a general idea of Daphne's architecture. Regarding the latter, the following is stated in the abstract of the said thesis:

> "This system has been developed using a layered architecture, where the first layer is a set of front-ends which are deployed on different client machines, including computers, virtual reality headsets, and physical embodiments. The next layer is a server which distributes the requests the user makes from all the front-ends to the different back-ends, which are the ones responsible for performing the basic functionalities of the system. A system of skills, pieces of software which provide the functionalities the end user uses, is built on top of this server. Finally, the back-ends and the skills use different data sources to perform their functionalities."

Other references with the latest additions and contributions to Daphne (mainly made by Antoni Virós himself) include [7], [8], [9] and [10].

Daphne's architecture proved to be extremely versatile, and as a consequence, it became the baseline for other projects carried at the SEAK Lab (namely, Daphne-EDL, which extends the design assistance to the Entry and Descent Landing phase; or Daphne-AD, which is a basic analysis tool for Anomaly Detection in generic datasets). In particular, Daphne-AT is one of the said "children" projects, and hence its name, since AT stands for the Anomaly Treatment approach (which includes not only anomaly detection, but also diagnosis and response).

It is worth mentioning Daphne's architecture because it has strongly determined the one for Daphne-AT. The details of the latter will be properly described in this thesis.

1. Introduction

1.1. Motivation and background

Long delays in communications between spacecraft and ground station are inherent to deep space exploration missions - for example, a round trip communication between Earth an Mars can take up to 40 minutes-. In this scenario, immediate mission control assistance might be unfeasible, and therefore, time-sensitive anomalies - like a fire or a pressure leak - become a critical obstacle towards mission success. Hence, improving the degree of autonomy of both spacecraft and crew in the context of anomaly treatment is a key milestone towards future missions to the Moon, Mars and beyond.

1.1.1. The Human Research Program (HRP) at NASA

The HRP was established in October 2005 at the Johnson Space Center (JSC) in response to NASA's decision to focus its research investment on investigating and mitigating the highest risks to astronaut health and performance in support of exploration missions. The main goal of the HRP is to provide human health and performance countermeasures, knowledge technologies and tools to enable safe, reliable, and productive human exploration. In particular, and regarding the content of this thesis, there are two specific objectives of the HRP that are worth remarking:

1. Developing capabilities and technologies in support of human space exploration that focus on mitigating the highest risks to crew health and performance.

2. Develop technolgies that serve to reduce medical and environmental risks and to ensure effective human-system integration across exploration mission systems.

To tackle and accomplish such goals and objectives, and to ensure a consistent, integrated process for managing human system risks that are critical to successful human exploration beyond low Earth orbit, the HRP has a detailed risk portfolio (which is publicly available at `https://humanresearchroadmap.nasa.gov/`). If the evidence base for a risk indicates gaps in knowledge about characterizing or mitigating the risk, such risk is identified as required to be researched by the HRP.

1.1.2. The HCAAM project

The development of the Daphne-AT tool presented in this thesis was triggered by the publication of NASA's Human Exploration Research Opportunities (HERO) in 2017. In such document, HRP solicited research in the topic of Human Capabilities Assessments for Autonomous Missions (HCAAM) Virtual NASA Specialized Center of Research (VNSCOR) (among other topics). The purpose of such topic is to focus research on providing the basis for the development of standards, guidelines and automation tools for an advanced autonomous decision-support system for use by NASA's astronaut population that will enable real-time human performance support for autonomous spaceflight.

This particular topic aims to address several risks from the mentioned HRP risk portfolio. However, the one that is most related to the tool described in this thesis is the "Risk of Inadequate Human-Computer Interaction (HCI)". Such risk entails the following HRP research needs:

- The need of HCI guidelines (e.g., display configuration, screen-navigation) to mitigate the performance decrements and operational conditions of long duration spaceflight.
- The need to understand how emerging multi-modal and adaptive display and control technologies are best applied to the design of HCI for proposed long-duration DRM (Design Reference Missions) operations.
- The need of guidelines to ensure crewmembers receive all of the information required to accomplish necessary tasks in a timely fashion, even when operating autonomously.
- The need to define the acceptable level of risk for HCI performance relative to terrestrial baselines.

1.1.3. HERA

Located at the JSC, the Human Exploration Research Analog (HERA) is a unique three-story habitat designed to serve as an analog for isolation, confinement, and remote conditions in exploration scenarios. Figure 1 and 2 consist of pictures of the exterior and interior of such analog respectively.

Periodically, NASA selects a crew of 4 astronauts to be confined in the HERA facilities for long periods of time, and then conduct research and studies including topics like behavioral health and performance assessments, communication and autonomy studies, human factors evaluations, and medical capabilities assessments.

Each of such confinements is referred as a “mission”, and missions are packed in “campaigns”. The first HERA campaign consisted of four 7-day missions, while the latest fifth campaign (currently ongoing while this document is being written) consists of four 45-day missions.

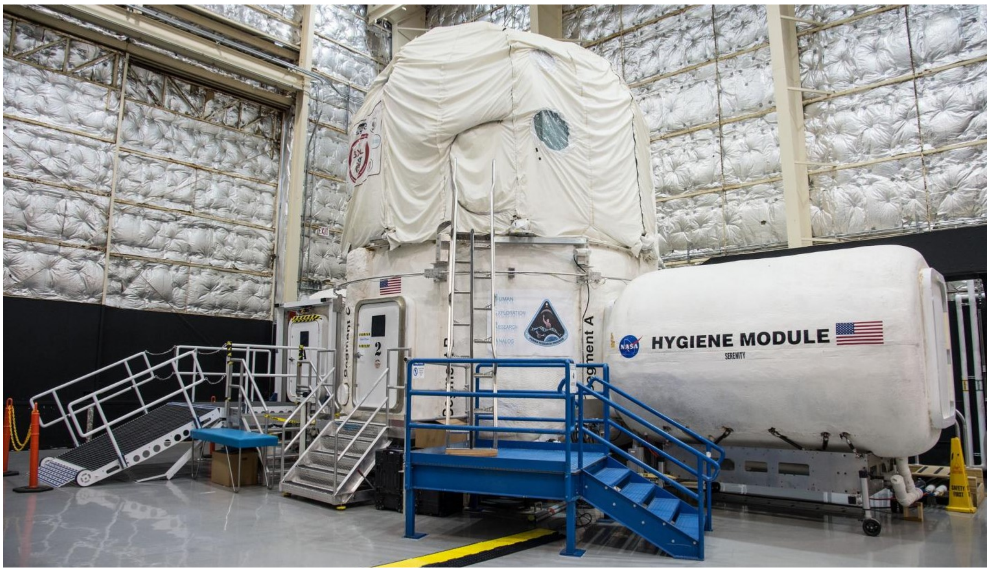

Figure 1: Picture of the exterior of the HERA analog.

Figure 2: A 360^{o} view of part of the interior of the HERA analog.

1.2. Problem statement and general objectives

Deep space exploration missions have large communication delays, and some on board anomalies require immediate attention to prevent mission failure. This is specially critical in manned missions, where the crew's life is at stake. Due to the said delays, mission control might not be able to help treating such anomalies. As a result:

The general objective of this thesis is to study on board automation tools as a means to increase the level of autonomy of both spacecraft and crew in the context of anomaly treatment.

1.3. Literature review and research gaps

Virtual assistants (VA) are intelligent agents designed to "augment human intellect" [14] and reduce human cognitive workload while performing some task. VAs typically provide a natural language interface, and often use some sort of cognitive architecture to model and anticipate user intention to facilitate the establishment of common ground.

VAs are now ubiquitous in our daily lives, as commercial programs such as Siri or Alexa help us organize our lives and perform mundane tasks. VAs have been studied for years in aircraft operations as a means to increase situational awareness (SA) and reduce cognitive workload (CW) during tedious or otherwise critical tasks [15], [16]. VAs have also been studied in space operational contexts to some extent [17].

There is, however - and to the best of our knowledge - a research gap regarding the design of a VA to support crew in the specific context of anomaly treatment. Perhaps the most relevant example of work in this area is the work by Michael Dorneich and Karen Feigh's groups, including Tokadli and Dorneich [18-20], who studied the use of VAs to support astronauts who are on missions where real time communication with Earth is not feasible to solve many of the problems that might arise. Tokadli and Dorneich conducted interviews with astronauts and developed a number of models such as Abstraction Hierarchies, Decision Action Diagrams, and a functionality matrix to define the cognitive functions performed by the crew, and how communication delays and interruptions may affect them. They also proposed a concept of operations and discussed the roles the VA could play in such a situation, including detection and diagnosis of anomalies, alerting crew and mission control of an emergency situation, offer and evaluate alternative courses of action, and simulating those courses of actions among others. Finally, they also proposed a set of design requirements for such VAs [18].

1.4. Specific objectives and scope of the thesis

A manned spacecraft consists of multiple integrated complex subsystems that function in synergy. The complexity of each subsystem makes the study of the anomaly treatment problem complicated even when such study is focused on a single one of them. Additionally, anomalies associated to different subsystems might require significantly distinct treatments.

To address the anomaly treatment problem in a thorough and complete manner, this thesis narrows its focus to anomalies that are somehow limited to the Environment Control and Life Support System (ECLSS) of a spacecraft.

There are several ways to increase the autonomy of spacecraft and/or crew during a space exploration mission, as for example, integrating decision-making automated systems on the spacecraft itself or increasing/broadening the crew's training. However, this thesis focuses on the use of a VA to increase such autonomy by exploiting the strengths of human-machine interaction.

Hence, the specific objectives of this thesis can be summarized as:

To develop a baseline version of a VA to help astronauts in the process of treating anomalies of the ECLSS system of the spacecraft.

To design a series of experiments to test the said VA and measure human performance enhancement in a high fidelity analog.

As introduced in the preface of this thesis, such VA will be referred as Daphne-AT onward in the document.

1.5. Scope and delimitation

Daphne-AT emerged as a response to the HERO publication, since NASA selected this proposal (together with 24 other and among a total of 100) as a part of its long term plans to lay the groundwork for future deep space exploration missions. Ultimately, the VA is meant to be tested at the HERA facilities in a series of experiments that will take place during the sixth campaign of the analog.

The project for the development of the tool is meant to be four years long, and it will be carried by an entire team at the SEAK Lab and in collaboration with NASA itself. However, this thesis is only a partial contribution to the project's earliest stage, and it has been submitted as a partial fulfilment of the requirements for a Final Bachelor's Thesis (undergraduate level). Therefore, it is worth to clearly define the scope of the thesis and to establish its delimitation.

What this thesis intends to accomplish:

- To identify the key features that the VA should have.
- To implement the software for the said baseline version of the VA.
- To carry pilot tests to debug the software and improve its robustness.
- To provide a preliminary design for the experiments at the HERA analog.

What this thesis does not intend to accomplish:

- To fully establish the basis for the development of guidelines and standards for automation tools for autonomous decision-support systems for NASA, as requested in the HERO 2017 document.
- To provide a fully operational and working version of the VA for "commercial use".
- To actually perform the experiments at HERA and present the corresponding results.

1.6. Overview and structure of the document

The approach of the document consists of providing a detailed description not only of the tool itself, but also of the development process and the motivations behind the decisions made.

The document is organized as follows: Section 2 describes the Graphical User Interface of the VA. Also, its main functionalities and the underlying architecture are presented. Section 3 gives detailed explanations on how the software that supports the tool has been implemented. Section 4 is dedicated to the design and execution of the pilot experiments. Additionally, a preliminary design of the experiments at the HERA analog is procured. Section 5 consists of a thorough analysis of the required future work for the project. Finally, the conclusions of the thesis are offered in Section 6.

2. Description of the tool

Introducing Daphne-AT can be a bit challenging without first seeing the actual VA in a life demonstration or a video. To compensate for that, the Graphical User Interface (GUI) of Daphne-AT is presented in detail at the beginning of this section. Then, its desired functionalities and architecture are described too.

2.1. The GUI

The GUI has been designed to provide a comprehensive and natural coverage of the anomaly treatment process. We decided to break down such process in three stages, which not only are relatively different in purpose, but also follow a chronological order most of the times. Such stages are:

1. *Detection*: The presence of an anomaly is identified. This stage would attempt to answer to the question "*What* is happening?"

2. *Diagnosis*: Possible explanations or causes of the anomaly are provided. This stage would attempt to answer to the question "*Why* is this happening?"

3. *Response*: A course of action is taken to resolve the anomaly. This stage would attempt to answer to the question "*How* do I prevent this from happening?"

Apart from that, and as further explained in Section 2.2, we want the VA to be able to communicate with the user in natural language. All these considerations lead to the GUI depicted in Figure 3. As it can be appreciated, the GUI consists of two main blocks:

- On the left side, the *anomaly treatment* block. This block allows a quick assessment of an anomalous scenario by displaying summarised information for each of the stages of the anomaly treatment process. From top to bottom, four more components can be discerned:
  - The *anomaly detection* window:
  - The *sensor data* window.
  - The *anomaly diagnosis* window.
  - The *anomaly response* window.

- On the right side, the *chat* window. This window allows to communicate with the VA by making questions or commands in natural language (either by manually typing them through the keyboard or using speech recognition). It is “detached” from the previous block in the sense that such requests can me made at any point of the anomaly treatment process, and they do not need to be related to the information displayed in such block.

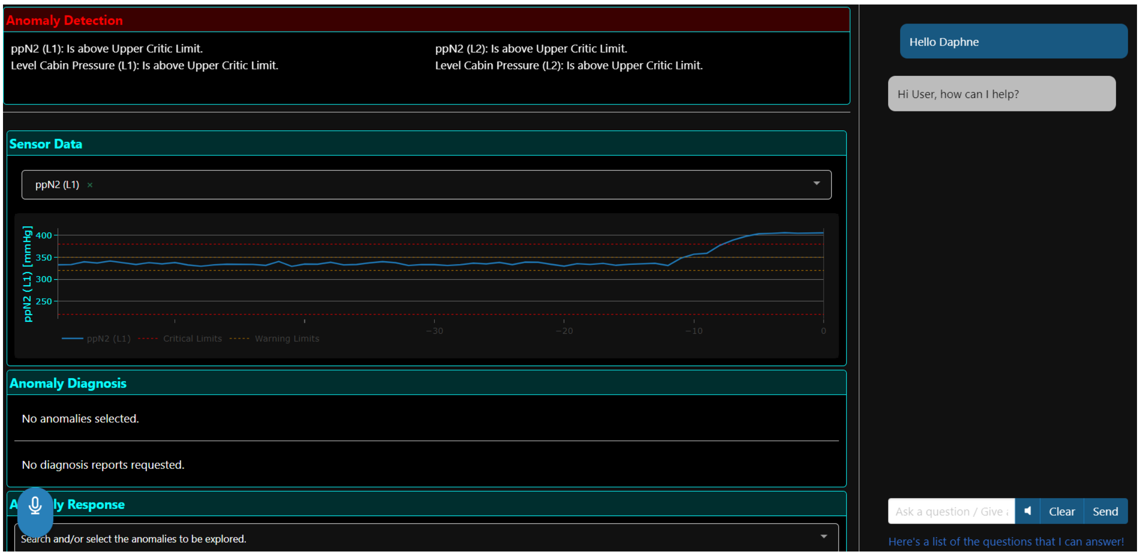

Figure 3: Overall display of Daphne-AT GUI.

2.1.1. The anomaly detection and sensor data windows

Figure 4 consists of a detailed view of the anomaly detection and sensor data windows.

The sensor data window allows the user to plot the evolution of any set of the spacecraft’s sensor readings by selecting them from the dropdown menu at the top of the window. When only one measurement is selected to be displayed, its warning and critic limits are shown too (which is the case depicted in Figure 4). If more than one measurement is selected simultaneously, such limits are not shown, to avoid confusing the user with too many lines in the plot.

The anomaly detection window aims to bring the attention to the user whenever a measurement acquires an anomalous value. For this purpose, it displays a list of all the measurements that are off nominal (specifying whether they are above/below a warning/critic limit) and it changes its color according to the most severe anomalous measurement (that is, red if any of the measurements has reached a critic limit, and orange otherwise). To ensure that the user never misses a new anomaly, this window is anchored to the top of the screen (and never moves when scrolling down). Additionally, an alarm is triggered when this window changes.

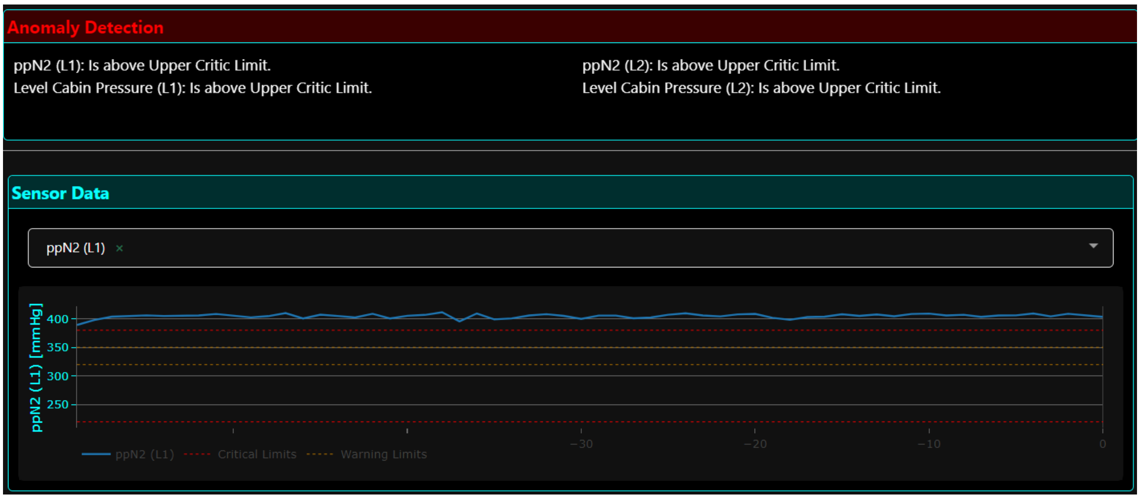

Figure 4: The anomaly detection and sensor data windows. The sensor data window is displaying the evolution of the “ppN2” measurement (as a solid blue line) together with its warning and critic limits (the orange and red dashed lines respectively).

The items on the list displayed in the anomaly detection window are clickable. When the user clicks on them, they become “selected” and appear in the anomaly diagnosis window.

2.1.2. The anomaly diagnosis and anomaly response windows

Figure 5 entails a detailed view of the anomaly diagnosis and anomaly response windows.

The anomaly diagnosis window allows the user to ask Daphne-AT to look for possible root causes of an anomaly. To do so, as mentioned before, the user can select a set of anomalous measurements from the anomaly detection window. Upon click, they appear on the upper slot of the anomaly diagnosis window. Then the user can click on the “Diagnose” button, and Daphne-AT will provide her answer on the lower slot.

As shown in Figure 5, the answer from Daphne-AT consists of two columns. The left one simply contains the set of measurements that the user had selected when calling the diagnosis functionality. The right one contains a list with the possible causes that might be producing such anomalous measurements. Daphne-AT also provides a score for each of the possible causes, according to how well does each cause explain or fit to the selected anomalous measurements.

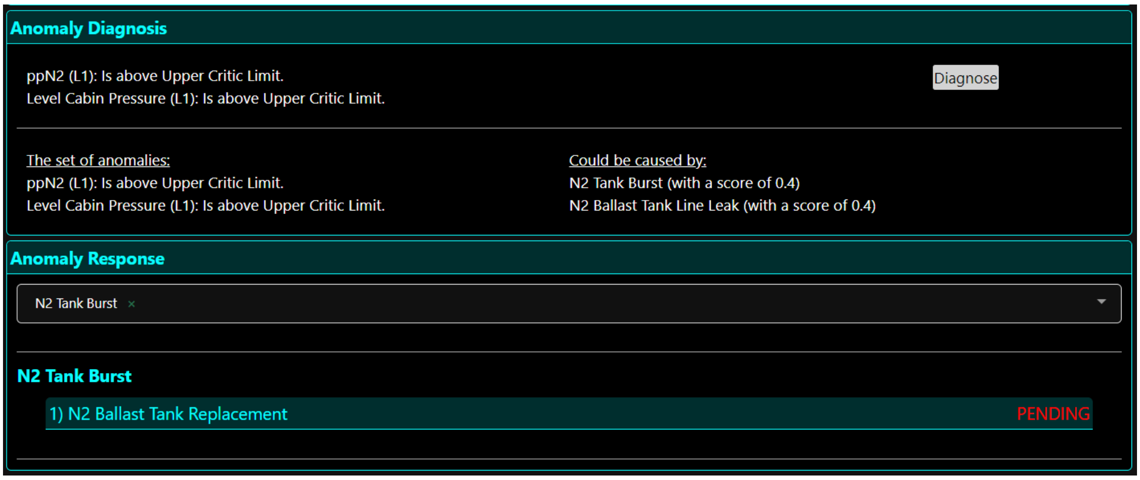

Figure 5: The anomaly diagnosis and anomaly response windows.

The items on the possible causes list are also clickable. When clicked, they become "selected" and appear in the anomaly response window.

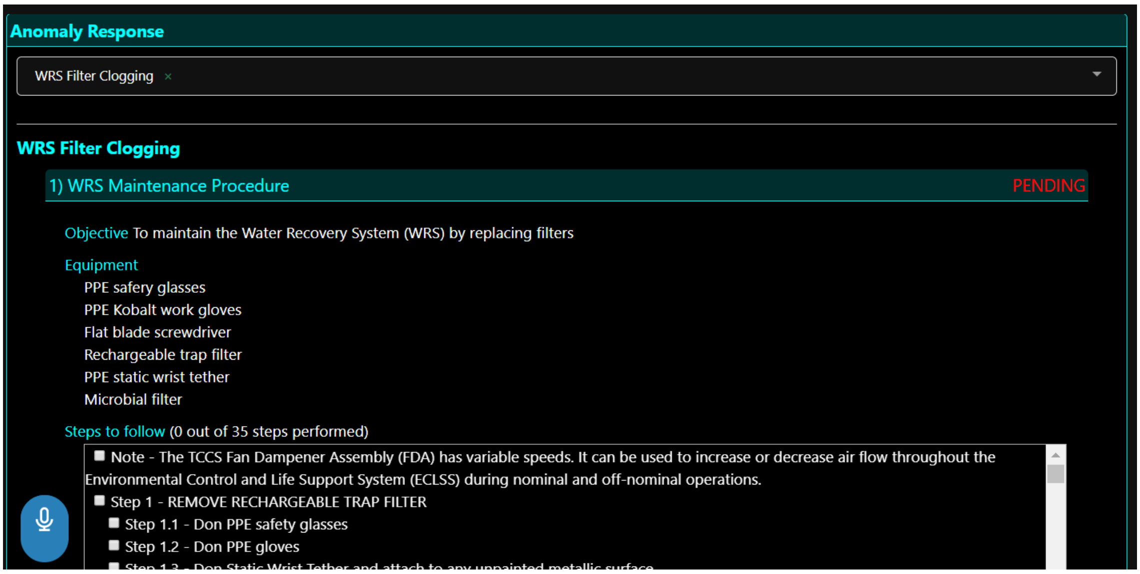

Figure 6: The anomaly response window, in detail.

The anomaly response window allows the user to explore the required procedures to solve an anomaly. To do so, the user can either select a possible anomaly cause from the suggestions of Daphne-AT or from the dropdown menu at the top of the anomaly response window (this option is offered since the user might disagree with Daphne-AT's suggestions). Once the anomaly root cause is selected, all its related procedures (there could be more than one) are displayed as a list. Each item of the procedure list is clickable, and when "unfolded", it displays all the information that such procedure contains, just as shown in Figure 6 (note how it differs from Figure 5).

As depicted in Figure 6, each procedure has three main pieces of information:

- *Objective*: The purpose of the procedure.

- *Equipment*: The required tools and devices to perform the procedure.

- *Steps*: A detailed breakdown of the steps that need to be followed in order to complete the procedure, as a scrollable box. Each step has a checkbox to its left, so that the user can check it upon completion. Once all of the steps are marked, the procedure status (at the top right corner of the slot) switches from "pending" to "completed".

2.1.3. The chat window

Figure 7 consists of a detailed view of the chat.

The chat allows the user to communicate with Daphne-AT in natural language. The user can send a question or command to Daphne either by typing it on the question bar at the bottom of the block or by using the speech recognition feature. To enable the latter, the user has to click the "microphone" button at the bottom left corner of the screen (see Figure 3).

In general, the questions that Daphne-AT can currently answer are about information regarding a specific anomaly or procedure (or how they relate) and the status of the sensor readings. A list of all the requests that Daphne-AT is currently able to answer is shown hereunder (Daphne-AT is able to handle them when phrased in other ways, to some extent):

- Show the current value of X measurement.
- What are the thresholds for measurement X?
- Check the status of X measurement.
- What are the potential risks of X anomaly?
- What are the signatures of anomaly X?
- What subsystems does anomaly X affect?
- What are the procedures for anomaly X?
- Provide procedure X.
- What components does procedure X impact?
- How long will it take to correct anomaly X?
- How long does it take to complete X procedure?

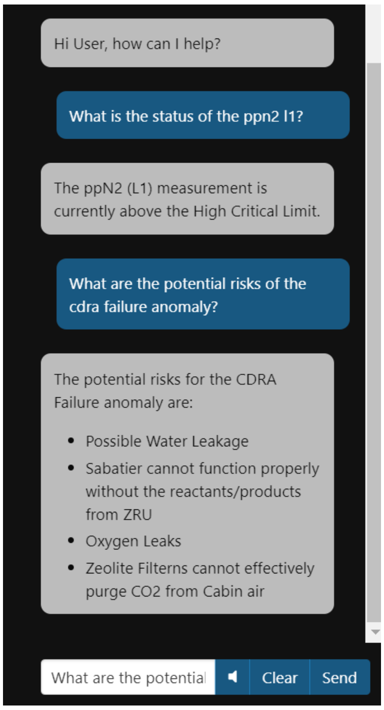

Figure 7: The chat display.

2.2. Desired functionalities

The GUI described in the previous section already provides an idea of what we want the VA to be able to do. In this section, a detailed breakdown and description of all its desired functionalities is provided [3]. Functions are prioritized following Kano's nomenclature: must have/should have/could have [11].

1. *Detect anomalies:* The VA must be able to perform continuous active monitoring of the environment and spacecraft subsystems and components, and identify problems or potential problems using real time data analysis. It should characterize the anomaly based on, for example, scope, urgency, and potential impact. It is important that the VA also predicts anomalies before they occur with some level of certainty and accuracy.

2. *Diagnose anomalies:* The VA must be able to help the crew identify the root cause of the anomaly. Specifically, it must able to leverage various data sources (historical anomaly database, flight rules, component data sheets) to infer likely root causes. It should also be able to provide multiple possible root causes ranked by likelihood.

3. *Recommend a course of action:* The VA must provide support to the crew for various tasks related to troubleshooting anomalies by suggesting users one or more recommended

course(s) of action. The VA should also run tests to identify the potential impacts of the decisions taken to resolve the anomaly, and ensure that those decisions do not create any future risks.

4. *Provide a natural language Interface:* The VA must have the ability to hold conversations with the user in natural language. It should be able to understand, analyze, and ask questions (for example, follow up questions during diagnosis) in human language.

5. *Prioritize anomalies:* In the event of simultaneously occurring anomalies, the VA should be able to prioritize them based on potential impact or risks.

6. *Relate anomalies:* The VA should be able to relate an anomalous event caused by another or two or more anomalies that likely have the same root cause.

7. *Support multiple users:* The VA should be able to support simultaneous use by multiple users. Each user should be able to have its own environment for the VA.

8. *Prepare summarized reports:* The VA should be able to generate event reports containing detailed anomaly detection, diagnosis and resolution procedures, and update historical databases whenever required.

9. *Provide explanations:* The VA should be able to provide explanations for all its decisions, actions and recommendations.

10. *Take initiative:* The VA should be able to take initiative in the dialogue with the user, as opposed to only reacting to a user's request.

11. *Convey its own limitations:* The VA should be able to characterize and communicate its own limitations and the level of confidence in all its decisions and recommendations.

12. *Adapt to different users:* The VA could adapt to individual user differences and preferences such as cognitive style, interaction mode, or level of initiative.

As stated at the beginning of this document, the tool is currently in an early development stage. Hence, not all the functionalities stated in the previous list have been implemented yet, but all of them have been mentioned for the sake of completeness of this thesis.

Up to the time of the writing of this document, the current version of the VA supports most of the features from the desired functionalities 1, 2, 3 and 4. It also supports 5 to some extent. Finally, although it still does not support multiple users, minor modifications would be needed to do so. The rest of the functionalities are not yet implemented at all, but we are certainly working towards them.

2.3. Architecture

Figure 8 illustrates the architecture of the VA. It consists of three main entities (grouped by purpose/functionality):

- The *User Interface*: This entity allows the VA to interact with a human. It receives the user queries and commands and provides the proper output in a comprehensive manner.
- The *Data Sources*: This entity stores and provides all the information that the VA needs to assess an anomalous situation. These include the source of the sensor data (a real time feed provided by a simulator) and the expert knowledge database (which contains information to help Daphne-AT to make decisions).
- The *Daphne-AT Brain*: This entity is in charge of "thinking", and it mainly acts as the link between the other two entities. Depending on the nature of the task to carry, a different piece of software (refered as *skill*) addresses the issue. There is one skill for each of the anomaly treatment process, that is, a *detection* skill, a *diagnosis* skill and a *response* skill.

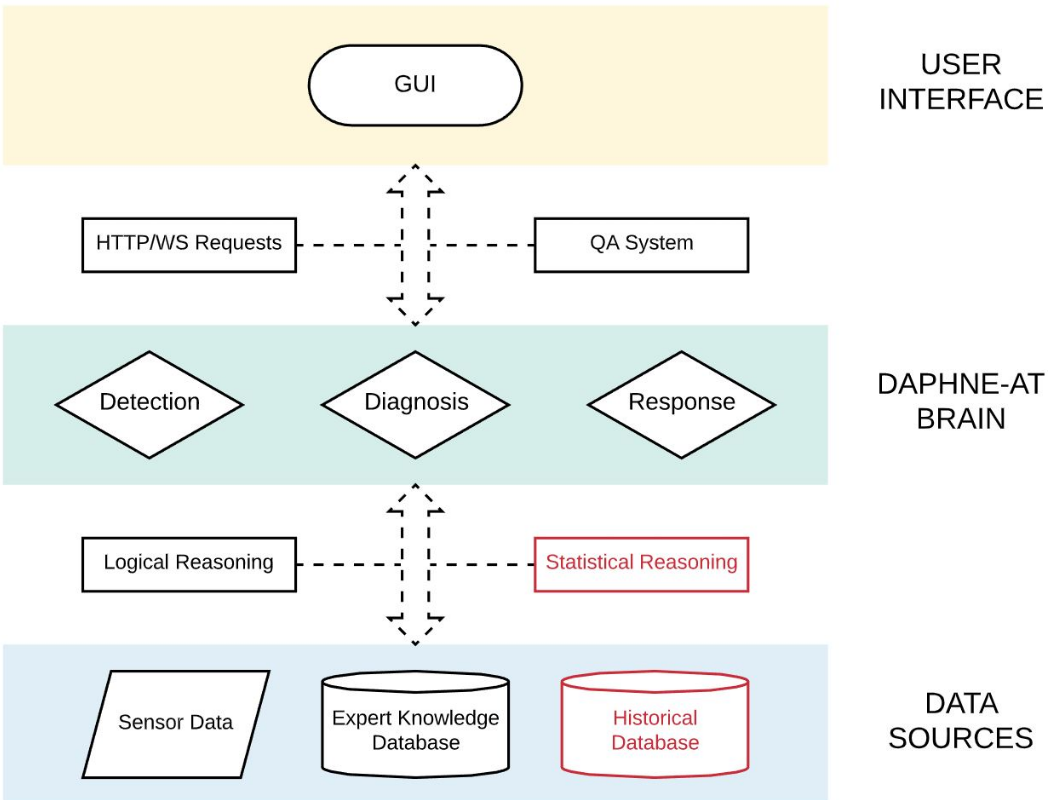

Figure 8: Architecture of Daphne-AT.

The schema on Figure 8 also depicts some of the relevant processes that take place when entities communicate between them. For the case of the GUI and the Daphne-AT brain, and bearing in mind that that this tool is deployed as a web server:

- *HTTP Requests*: This protocol is used mainly to handle processes that are intentionally triggered by the user (such as, for example, requesting the diagnosis of a set of detected anomalous measurements).

- *WS (websockets) Requests*: This protocol is used to handle generally automated (and asynchronous) processes that the user is not necessarily aware of (like the constant update of the telemetry feed plot).

- *QA system*: Although the handling of a user command is triggered by an HTTPS Request from the GUI, the command goes through an exhaustive Natural Language Processing (NPL) mechanism before the proper skill actually addresses the topic, and this is why it deserves to be mentioned separately. The details of this system are presented in Section 3.5.

Regarding how the Daphne-AT brain and the Data Sources are linked, it is first necessary to note that there are a couple of items marked in red in the schema of Figure 8: the historical database data source and the statistical reasoning feature. This is because we believe that implementing such items is key for the development of the VA, but we do not have the resources to do so yet. This issue is further explained in Section 5 of this document.

Because of the latter, Daphne-AT is mostly knowledge-based right now, and that is what the *logical reasoning* feature refers to. That is, all her knowledge and "thinking" abilities are basically either hard coded or implicit in the expert knowledge database structure (Section 3.3).

The main advantage of using a knowledge-based approach is that it is extremely reliable when used in known situations - that is, situations that have been studied beforehand-. In contrast, this approach is quite blind to new scenarios, and is likely to fail when facing a situation that has not been characterised in advance. In the particular case of Daphne-AT, this means that the more anomalies are studied, the more rules need to be hard coded, and that Daphne-AT might fail to assess a novel anomaly scenario correctly. Hence, the rule-based approach does not scale well to the size of the problem.

To solve this flaw, we want to study the possibility of implementing a data driven approach in the future. This is why we included the *statistical reasoning* feature in the schema presented in Figure 8.

3. Implementation of the tool

The code for Daphne and all its children projects can be found at `https://github.com/seakers` . In particular, for the Daphne-AT tool, the back end and front end codes can be found at the `https://github.com/seakers/daphne_brain` and `https://github.com/seakers/daphne-at-interface` repositeries respectively. For the first one, the Django web framework was used, while the latter was implemented in Vuex.

In this section, the implementation of the most relevant pieces of software is presented. First, the simulator that provides the sensor data and the software to handle it is explained. After that, the knowledge graph, the diagnosis skill and the QA system are described in detail.

3.1. The simulator

Daphne-AT is fed with the real time sensor data readings from the Environment Control and Life Support System (ECLSS) of the spacecraft or HERA. This allows the VA to monitor the health of such system and detect possible anomalies. It is worth mentioning though that this is not the only way to detect anomalies, since particular events - like a minor liquid leakage from a pipe - might not manifest in such sensor data readings, but could be detected by the crew itself, which could trigger then the anomaly treatment process by asking Daphne-AT about the issue.

The source of such sensor data readings will not be real, and hence, a piece of software that simulates it is required. For the experiments at HERA, and in the near future, a simulator provided by NASA will be used for such purpose. Onward in this document, we will refer to such simulator as sECLSS.

The sECLSS is rather simple. One can simulate its own custom ECLSS scenario by performing the following tasks:

1. *Defining the involved measurements*

   All the possible parameters that the ECLSS entails and that could be read by a sensor are determined (to be referred as "measurements"). For each measurement, a name, parameter group, units and nominal value, standard deviation factor and proper warning/critic limits have to be established. For example, a measurement that depicts the lectures obtained by an ammeter monitoring the fuel cells of the spacecraft could be defined as:

- *Name*: Fuel Cell Current
- *Parameter group*: Subfloor
- *Units*: A (Amperes)
- *Nominal value*: 15
- *Standard deviation factor*: 10
- *Lower Critic Limit (LCL)*: 12
- *Lower Warning Limit (LWL)*: 14
- *High Warning Limit (HWL)*: 16
- *High Critic Limit (HCL)*: 18

Where the standard deviation factor is used to introduce some noise in the simulated value of the measurement. If NV stands for the nominal value of the measuremnent and K for the standard deviation factor, then the the standard deviation *std* for such noise is defined as $std = NV/K$.

2. *Defining the possible anomalies*

An anomaly is "designed" by determining a certain way in which the ECLSS could fail, and then characterising how such failure would affect the simulated measurements. We will refer to this as the "signature" of the anomaly. Then, for each anomaly, a name, a delay time lapse and a set of affected measurements have to be defined. After that, for each affected measurement, a phase in and phase out slope and their corresponding target values are set. For example, a partial failure of a Fuel Cell of the spacecraft (produced perhaps by overheating) could be defined as:

- *Name*: Fuel Cell Overheating
- *Delay*: 10 (in seconds)
- *Affected measurements*: Fuel Cell Current
  - Phase in slope: -0.1
  - Target in value: 13
  - Phase out slope: 0.01
  - Target out value: 15

This means that, 10 seconds after the Fuel Cell Overheating anomaly starts occurring, the Fuel Cell Current will drop from its nominal value to a target value of 13 [A] (which, from the previous definition of the measurement, falls between its LCL and LWL) at a pace of 0.1 amperes per second (negative slope). Then, when the anomaly is resolved, the Fuel Cell Current will increase again at a pace of 0.01 amperes per second until it reaches a target value of 15 amperes (which, from the previous definition of the measurement, corresponds to its nominal value). Although in this example is not depicted, a single anomaly can affect more than one measurement simultaneously.

3. *Executing the simulation*

   The anomalies are manually introduced and removed from the nominal status of the spacecraft. That is, once the simulation is started, the sECLSS outputs a set of simulated values for all the defined measurements. Such values oscillate around the nominal values of the measurements, according to its definition. Then, an anomaly can be manually phased in and out, and the output will vary according to the definition of the anomaly (increasing or decreasing the affected measurements only).

This simulator is far from being realistic. It is not model based, meaning that all the physical phenomena that describe a real ECLSS environment are completely neglected. Also, the only perturbations that can be introduced are linear increases and decreases of the measurements' values, and such perturbations are completely independent between them. That being said, this software was provided by NASA, and it is the software that will be used in the HERA habitat experiments. The development of our VA was adapted to and limited by this situation.

Figure 9: NASA's ECLSS crew screen.

All the above described tasks can be performed from a "special user screen" that the sECLSS is equipped with, the details of which fall beyond the purpose of this text. However, the sECLSS also has a "user screen" or "crew screen", which is the one that is shown to the crew in the HERA habitat, and it mimics the display that would be available in a real spacecraft. Such screen is depicted in Figure 9.

After seeing the ECLSS crew display in Figure 9, it is worth making some comments about a couple topics which have already been introduced in this thesis.

First, the reader might notice that the overall design of the GUI of our VA (depicted in Figure 3) is very similar to the ECLSS crew screen. We did this in purpose and accounting for the human factor. The HERA crew will already be familiar with the ECLSS screen display and color codes, and hence, it is highly convenient for us to make the integration of our VA as smooth as possible for them.

Second, if the reader was wondering what the "parameter group" field in each measurement's definition stood for, now it becomes clear. The measurements are packed in the ECLSS screen according to its functionality or location in the HERA habitat. In particular, all the available parameter groups are the following:

- *L1 and L2*: These groups - one for each floor of the habitat - comprise the measurements that describe the overall status of the habitable part of the analog (as for example, the cabin air's temperature or pressure).

- *Subfloor*: This group entails the measurements that relate to the subsystems located under the floor panels of the analog (like batteries or pipes).

- *Crew*: The measurements in this category refer to the measurements that are individual to each crew member (such as radiation dose).

- *Trace contaminants*: This one is pretty self descriptive. It contains all the measurements that correspond to the concentration of some undesired and residual atmosphere component.

3.2. The anomaly treatment process

Daphne-AT has one essential difference from the other "Daphne tools", and it is precisely that it has to monitor a bunch of data in real time (the sensor data provided by the simulator).

All the other "Daphne tools", like Daphne-EOSS (a VA that acts as a peer during the process of designing a satellite constellation), lack this “real time paradigm”. The information that they provide is purely user reactive, that is, the VA performs actions upon request. There is one feature though that was recently incorporated to the Daphne-EOSS tool that somehow resembles this “real time paradigm”. In its latest version, such tool looks on its own for new suitable satellite constellations (according to the problem that the user is trying to solve), and makes suggestions in an active manner (without a user’s request). To do so, an independent thread is deployed in the back end, and a genetic algorithm is run to perform such search.

This idea of deploying an independent thread to perform a parallel task is exploited for the Daphne-AT case. A system of three threads is in charge of monitoring the real time feed provided by the sensor data simulator. The purpose of two of such threads is described below, while the third one will be explained later on.

- *The simulator thread*

  The purpose of this thread is to host the sensor data simulator.

  In case such simulator is provided by a third party software, the only task of this thread is to properly receive and parse its output. This is the current case for Daphne-AT, since we are using NASA’s sECLSS simulator. The sECLSS posts the simulated values to a cerain URL, so this thread is in charge of receiving the post and parsing the format of the data to the one that Daphne-AT uses.

  In the future, we would like to build our own sensor data simulator, as described in the future work for this project (see Section 5). In this case, the simulator thread would be used to host such software, and it would provide Daphne-AT with the simulated values periodically. In fact, the sECLSS software was not available to us from the very beginning of the project, so we coded a simulator that somehow had the exact same features than the actual sECLSS, and such simulator was hosted in this thread.

- *The Anomaly Detection thread*

  The purpose of this thread is to process the output from the simulator and detect anomalies within the simulated sensor data values.

  To do so, Daphne-AT loops over all the received measurements and checks if they are within its nominal limits (that is, between the warning thresholds). In case they are not, a message with the name of the anomalous measurement and the limit that is surpassing (whether it is above/below the warning/critic limit) is produced and sent to the front end to be displayed in the GUI.

The simulator thread needs to communicate with the Anomaly Detection thread to provide the simulated values, and the Anomaly Detection thread needs to communicate with the main thread of the code to send its output to the front end. The best way to do this is to use global queues, that is, queue objects that are available to all of the threads.

In particular, each pair of threads that have to communicate need two queues: one for each communication direction. If, for some reason, more threads were to be added in the future (to perform other tasks related to the anomaly treatment process), the amount of needed queues would increase as $\mathcal{O}(n^2)$, since a dense undirected graph has exactly $n(n-1)/2$ edges (here n stands for the number of threads). Also, each thread would have to deal with exactly $2(n-1)$ queues. Not only this does not scale well to the amount of threads, but it also is a coding nightmare.

To avoid the latter issue, the third thread (named *hub thread*) is introduced. The only purpose of this thread is to handle all the communications between threads. If a thread A needs to communicate with a thread B, then A sends a message to the hub thread, and the hub thread rebounds it to thread B. With this strategy, an additional thread is needed, but only $2n$ queues are required, and each thread (except the hub thread) has to deal only with two queues (one for the input information and another for the output). From the coding point of view, this design is way more modular, and it scales better to the amount of used threads.

Figure 10 consists of a schema of this thread system if two more threads were to be added. The entities coloured in yellow stand for items that have not been implemented yet. The solid double arrows stand for the communication queues. The dashed single arrows stand for the "information flow". In this scenario, the usefulness of the hub thread becomes clearer.

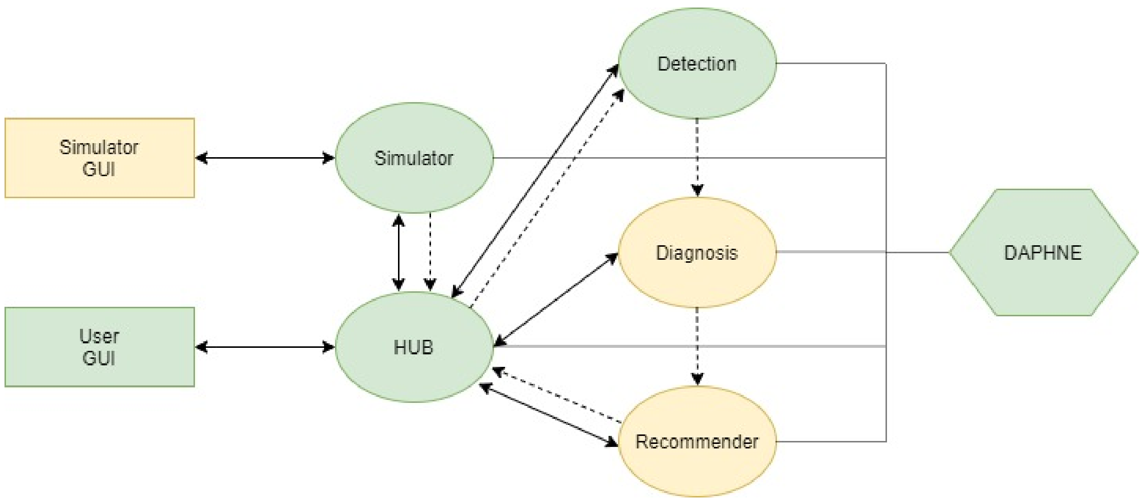

Figure 10: Schema of the anomaly treatment routine thread system.

Using queues to communicate threads has a great side advantage, which is that it provides an easy way to make sure that Daphne-AT never "falls behind" the simulator output.

Suppose that, starting at $t = 0$, the simulator produces the outputs $\{s_0, s_1, ..., s_k, ...\}$ separated by a time lapse of $\delta > 0$. If such outputs were to be processed sequentially, without skipping any, then the anomaly detection thread would have to process each output s_k in less than δ units of time, or it would lag otherwise. Effectively, if each s_k took $\delta + \varepsilon$ to be processed, with $\varepsilon > 0$, then the simulation thread would produce an output s_k at time $t = k\delta$, while its corresponding anomaly detection message would be obtained at $t' = k(\delta + \varepsilon)$, resulting in a time lag of $\Delta t = t' - t = k\varepsilon$, which increases linearly over time. Instead of that, if the anomaly detection thread disregards all the outputs from the simulator but the latest, then such lag is avoided, at the cost of, perhaps, not processing some of the incoming messages, which is a fair trade off for this specific application. In particular, the amount of skipped outputs would be, approximately, $N = \lceil(\delta + \varepsilon)/\delta\rceil = 1 + \lceil\varepsilon/\delta\rceil$ (time lapse required to process an output divided by the time lapse between outputs). Choosing some realistic values, like $\delta = 1$ $[s]$ and $\delta + \varepsilon = 3$ $[s]$ (which is a very loose margin to process an output s_k), Daphne-AT would be able to provide an anomaly detection assessment every 3 $[s]$, skipping only 3 out of 4 simulator outputs with a lag of, at most, 3 $[s]$. This situation is fair more desirable than the lagged scenario.

Since the current version of the VA only performs a simple threshold check, the lagged scenario described in the preceding paragraph is not likely to happen. However, we have already implemented fancier detection algorithms (such as SARIMA models and density-based methods, among other [2]) that are way more computationally expensive than the threshold check, and we intend to integrate them in the tool in the near future. If the machine in which the Daphne-AT is not fast enough, then such lag is likely to occur.

3.3. The knowledge graph

In Section 2.3, we presented the architecture of the VA, and we introduced the *expert knowledge* database. We chose a graph structure to support this resource, and this is why we refer to it as the *knowledge graph*. In particular, we implemented such graph using the neo4j framework. The main features that make such framework appealing to us are the following:

- *Graph structure*: Such a structure allows to group information in "chunks" (nodes) and establish relationships between them (edges). In particular, neo4j allows to define types of nodes and add as many labels or attributes as desired to them.

- *Friendly graphical interface*: The neo4j graph - which is hosted in a server - supports a very intuitive graphical interface, which allows to visualize the main information of the graph in a comprehensive manner, as depicted in any of the figures of this section.

- *Flexible query language*: The graph database can be queried using the cypher language, which is simple but very versatile.

The best way to introduce how the graph looks is, perhaps, by showing the structure of an anomaly node. Such node is depicted in Figure 11.

Figure 11: An isolated anomaly node in the neo4j graph.

In Figure 11, five types of nodes are shown:

- *Anomaly*: A node type whose only attribute is a string with the anomaly name.
- *Measurement*: A node that entails all the information of a measurement in the form of labels - like its name, parameter group, warning and critic limits, units, etc.
- *Procedure*: A node type the name of a procedure and a label with the estimated time to complete it.

- *Subsystem*: A node type with the name of a subsystem and a label with the main system that it belongs to.

- *Risk*: A node type containing a brief explanation of a certain risk.

How the nodes are connected is as important as the information entailed in the nodes themselves. For instance, in the above example, an anomaly node is tied to some risk nodes by the relationship type *can cause*, or to a particular subsystem by a *affects* type. This allows an intuitive navigation and querying of the graph.

To summarize, the reader might agree that Figure 11 is a concise and appropriate characterisation of a particular ECLSS anomaly by itself, and hence, it clearly supports our choice of the neo4j framework.

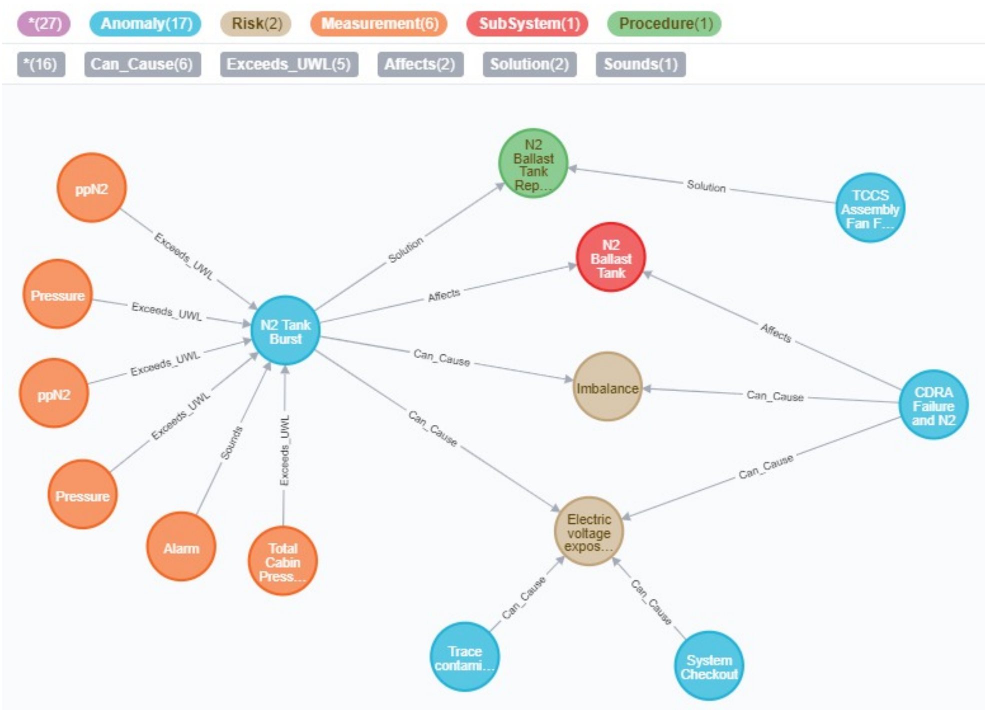

Figure 12: A close up display of various types of nodes and their relations in the neo4j graph.

Bear in mind though that the knowledge graph is not a set of isolated anomalies with some appended nodes, and that the anomaly nodes are not "the core" of the graph. In fact, almost all of the nodes in a graph are somehow related, and most anomalies share information to some extent, as depicted in Figure 12.

For instance, imagine that a member of the crew detects some malfunction within the N2 Ballast Tank subsystem (the red node in Figure 12). In an attempt to understand what is happening, the astronaut could ask Daphne-AT “What anomalies affect the N2 Ballast Tank?”. Daphne would retrieve all the anomalies (some of the blue nodes in Figure 12) that are connected to the red node by the *affects* relationship. Then the astronaut could ask Daphne-AT about the signature of each anomaly, and Daphne would provide a list of the measurements (the orange nodes in Figure 12) that are tied to each anomaly by any kind of relationship. Finally, the astronaut could assess what is the true cause of the anomaly by comparing the said lists with the status of the Daphne-AT/ECLSS screen and choosing the closest match.

The latter example not only shows how useful the graph structure is, but also illustrates the previous claim about the anomaly nodes not being “the core” of the graph. In fact, in this example, we “accessed” the graph via a risk node and then navigated to the measurement ones.

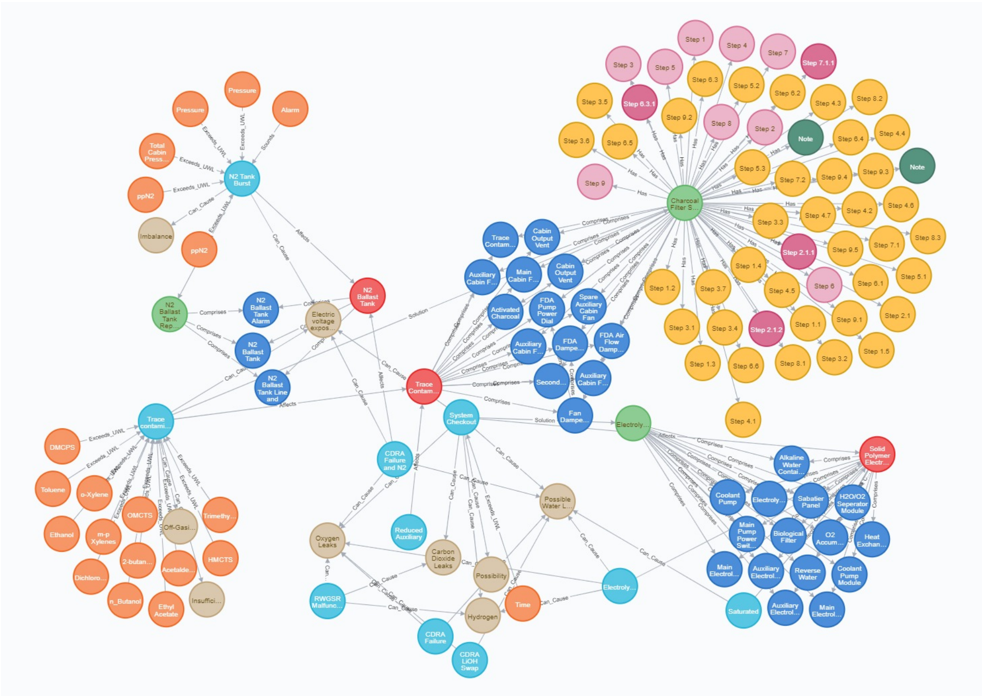

Figure 13: A detailed view of (part) of the neo4j graph.

Most of the information in the graph was retrieved or built from procedures that NASA provided us with. In general, such procedures aimed to perform some routine maintenance or emergence replacement of a subsystem of the HERA analog. Our effort to reverse engineer these procedures ultimately lead to a rather complete and dense graph, as Figure 13 attempts to illustrate.

3.4. The diagnosis skill

In this section, the current implementation of the diagnosis skill (and how did we get to it) is explained. To do so, some nomenclature shall be introduced first:

- Let $\mathcal{A} = \{a_1, a_2, ..., a_N\} \neq \emptyset$ be the set of all possible root causes of an anomaly. For example, a_1 could correspond to a *N2 Tank Burst*.
- Let $\mathcal{S} = \{s_1, s_2, ..., s_M\} \neq \emptyset$ be the set of all possible anomalous measurements. For example, s_1 could be *N2 exceeds the Upper Warning Limit*.
- For $1 \leq k \leq N$, let $\mathcal{S}_k \subset \mathcal{S}$ be the signature of anomaly a_k. To alleviate the nomenclature through this section, we will refer to $\mathcal{S}_k$ as the *symptoms* of anomaly a_k.
- Let $X \subset \mathcal{S}$ (that is, $X \in \mathcal{P}(\mathcal{S})$) be the set of anomalous measurements that the user is requesting a diagnosis for (as explained during the presentation of the GUI in Section 2.1).

We define the diagnosis function f as a function that assigns a set of possible anomalies to the set X of anomalous measurements that the user is requesting a diagnosis for, that is:

$$\begin{aligned} f : \mathcal{P}(\mathcal{S}) &\rightarrow \mathcal{P}(\mathcal{A}) \\ X &\mapsto Y = f(X) \end{aligned}$$

Our first approach was to define the image of X via f as the set of anomalies such that X was a subset of the anomaly symptoms, that is:

$$Y = f(X) := \{a_k \in \mathcal{A} : X \subset \mathcal{S}_k\}$$

This was quite robust, since the user can only request a diagnosis for anomalous measurements that are actually occurring, and therefore, the anomalies that were indeed happening would always appear in the answer provided by Daphne-AT. However, at some point we decided that we would include anomalous scenarios in which two anomalies would happen simultaneously. As the reader can easily check, if the occurring anomalies have non-common symptoms, then f would not include them in Daphne-AT's answer, even when they are indeed happening.

To solve this, the second approach we took was to define the image of X via f as the set of anomalies such that X has non-empty intersection with the anomaly symptoms, that is:

$$Y = f(X) := \{a_k \in \mathcal{A} : X \cap \mathcal{S}_k \neq \emptyset\}$$

This solved the issue for scenarios with combined anomalies, but the answer from Daphne-AT was always too vague. Since the diagnosis condition is too weak, the image set Y had always too many anomalies to be useful for the user.

Our final approach (the one that is currently implemented) is to use the previous definition of f, but adding some criteria to rank the anomalies in Y and cast the output down to a top K list. This criteria is in fact the score that Daphne-AT provides for each anomaly when a diagnosis is requested (see Figure 5).

Let $Y = f(X) = \{b_1, b_2, ..., b_P\}$, and note that, from how the diagnosis feature has been implemented in the GUI, the user is never allowed to request a diagnosis for an empty set (hence $X \neq \emptyset$). Now, consider the following functions:

$$\begin{aligned} t : Y \in Im(f) &\to (0,1] \subset \mathbb{R} \\ b_k &\mapsto t(b_k) := \frac{\#(X \cap S_k)}{\#X} \end{aligned}$$

$$\begin{aligned} r : Y \in Im(f) &\to (0,1] \subset \mathbb{R} \\ b_k &\mapsto r(b_k) := \frac{\#(X \cap S_k)}{\#(X \cup S_k)} \end{aligned}$$

Note that both functions are well defined since $X \neq \emptyset$ and thus $0 < \#X \leq \#(X \cup S_k)$.

Function t provides high scores to anomalies that could produce all the symptoms that are occurring on their own. However, it is biased towards anomalies with a lot of symptoms, since they are more likely to include the set X. Function r attempts to compensate for that by providing low scores if the anomaly also has symptoms that are not actually occurring.

Finally, the score function s is simply defined as:

$$\begin{aligned} s : Y \in Im(f) &\to (0,1] \subset \mathbb{R} \\ b_k &\mapsto s(b_k) := t(b_k) \cdot r(b_k) \end{aligned}$$

The closest to one $s(b_k)$ is, the better anomaly b_k explains the anomalous measurements that are occurring.

3.5. The question answering System

To abbreviate, we refer to all the the code that supports the natural language interface and the question answering feature as *QA system*.

(*Note*: Strictly speaking, the user can make requests to Daphne-AT not only in a question format, but also as commands (i. e., the user could either say “What is the procedure to solve X anomaly?” or “Provide the procedure to solve X anomaly.”). Through this section, in order to alleviate the nomenclature, we use the word “question” to refer to either of these types of requests that the user can make to the VA.)

The QA system mainly relies on two technologies:

- *Natural Language Processing (NLP)*: As its name indicates, it consists of a set of computational techniques that aim to analyze and synthesize human natural language (either text or speech). For this purpose, the spaCy library is used [12].

- *Classification algorithms*: Two Neural Networks (NN) are used to identify which task is the user requiring Daphne-AT to perform. The tensorflow's keras library is utilized [13].

In Section 2.1.3, a list of the questions that Daphne-AT is currently able to answer was presented. Adding a question to such list involves three distinguished stages:

1. *Question definition and formulation*

2. *Training*

3. *Execution*

This process is template based. That is, adding a question only requires building two files (a *.txt* to train the NN and a *.json* file to properly build the Daphne-AT answer) and coding a specific function to perform the task(s) that the question involves.

The main advantage of this approach is that adding a single question is quite straightforward and does not require a deep understanding of the QA system code nor the involved technologies (a coder with no idea about NLP or NN's can easily do it). In contrast, it does not scale well to the amount of questions. For example, a small modification in how the front end deals with the question output format would require reviewing all the *.json* files one by one.

The tasks that each of the above mentioned stages entail are detailed below together with an illustrative example.

1. *Question definition and formulation*

   (a) Define the task to be performed.
   e. g.: Display the thresholds of X measurement.

   (b) Identify the parameters of the question and the related skill.
   e. g.: Parameters: *measurement name name* and *parameter group*, Skill: *detection*

   (c) Think of all the possible formulations of the question in NL.
   e. g.: *Show me measurement X thresholds.*, *Can you provide the thresholds for measurement X?*, *What are the limits of X?*

   (d) Assign a four digit tag code to the question.
   e. g.: 1001 ($A - BBB$, where A points to the skill and BBB is a unique label for each question related to that skill).

This stage ultimately results in a question template in the form of the mentioned *.txt* file. An example is provided in Figure 14. In such figure, the first block stands for the number of question instances to generate for the training stage. The second block contains the type and name of the parameters of the question. The bit code in the third block indicates the skill that this question corresponds to (in this case, detection). The fourth block consists of a list with possible phrasings of the question.

1001.txt

```
1000
--
measurement measurement
parameter_group parameter_group
--
100
--
Show the thresholds of measurement ${measurement} ${parameter_group}.
Show the ${measurement} ${parameter_group} thresholds.
Provide the ${measurement} ${parameter_group} limits.
Provide the thresholds of measurement ${measurement} ${parameter_group}.
Provide the ${measurement} ${parameter_group} measurement limits.
What are the measurement ${measurement} ${parameter_group} thresholds?
What are the ${measurement} ${parameter_group} thresholds?
${measurement} ${parameter_group} thresholds.
${measurement} ${parameter_group} limits.
```

Figure 14: A *.txt* file with a question template for the training stage.

2. *Training*

   (a) Generate a training set by randomly picking question formulations from the *.txt* file and substituting the parametrization by real parameter values.
   e. g.: *Show me measurement **pressure** thresholds.*, *Can you provide the thresholds for measurement **ppN2**?*

   (b) Train each of the two neural networks to:
      i. Identify the skill to be addressed (first NN).
      ii. Identify the associated tag code (second NN), once the skill has been identified.

   The idea behind using two NN is that questions that address a certain skill are quite similar, but significantly distinguishable from the ones that target other skills. Although this proved to work better than a single NN for the Daphne tool, it is arguable that it brings any benefit to Daphne-AT, since questions that aim to different skills are still very similar. However, the two NN have been kept, as both tools share the QA system, and this approach is not detrimental for Daphne-AT's performance.

3. *Execution*

For each question, the said *.json* script template is built (an example is provided in Figure 15), and named with the question tag (i. e.: *ABBB*.json). When the user makes a NL question (either using the voice recognition feature or by typing it on the question bar):

   (a) The command is processed using the NLP spaCy library.

   (b) A list of predictions of the corresponding question tags is obtained (with their associated likelihoods).

   (c) The proper .json script is retrieved (the one with the same question tag) and an action is performed according to the information embedded in such file.
      i. Daphne-AT looks for the parameters that the question should contain as indicated in the *.json* file (lines 3-19 in Figure 15), and attempts to extract them.
      ii. Daphne-AT executes the specific function associated to the question task (line 23 in Figure 15).
      iii. Daphne-AT embeds its answer in an object to be interpreted by the front end. Such object generally consists of some sort of text (either raw or as a list), but can also include plots and images (lines 36-51 in Figure 15).

1001.json

```json
{
  "type": "run_function",
  "params":
  [
    {
      "name": "measurement",
      "type": "measurement",
      "options": "",
      "from_context": false,
      "mandatory": true
    },
    {
      "name": "parameter_group",
      "type": "parameter_group",
      "options": "",
      "from_context": false,
      "mandatory": true
    }
  ],
  "objective": "provide the thresholds for a certain measurement",
  "function":
  {
    "run_template": "dialogue_functions.get_measurement_thresholds<
                        data['measurement'], data['parameter_group']>",
    "results": [
      {
        "result_type": "list",
        "result_fields": {
          "threshold_type": "item['threshold_type']",
          "threshold_value": "item['threshold_value']",
          "threshold_units": "item['threshold_units']"
        }
      }
    ]
  },
  "voice_response": [
    {
      "type": "list",
      "begin": "The thresholds for the ${measurement} <${parameter_group}>
                    measurement are:",
      "repeat": "${threshold_type}: ${threshold_value} ${threshold_units}",
      "end": "."
    }
  ],
  "visual_response": [
    {
      "type": "list",
      "begin": "The thresholds for the ${measurement} <${parameter_group}>
                    measurement are:",
      "item_template": "${threshold_type}: ${threshold_value} ${threshold_units}"
    }
  ]
}
```

Figure 15: A *.json* file with a question template for the execution stage.

4. The experiments

As mentioned in the introduction, the tool described in this document is meant to be tested at NASA's HERA analog. However, before that, we have conducted a series of pilot experiments at our lab in order to test the tool in advance. The set up for both experiments, the corresponding procedures and some results for the pilot experiments are presented in this section.

4.1. HERA experiments

The mission during which the VA will be tested in the sixth HERA campaign is 45 days long, that is, six and half weeks. The crew of the mission will consist of four members.

The schedule for the mission is divided in two phases. The sessions in phase 1 are planned, which means that the subjects will know beforehand that they have an allocated slot to perform our experiment. In contrast, for phase 2, the subjects will not know when the experiment is going to happen, thus "catching them off-guard" and simulating a more realistic anomalous scenario. Also, phase 1 sessions will be individual, while phase 2 sessions will be in group.

During phase 1, we have been allotted with two one hour long sessions per subject and week, which adds up to 6 sessions per subject during the whole phase, and 24 sessions in total. We will use one of the sessions to expose the subject to an anomalous scenario with the assistance of Daphne-AT, and the other without.

For phase 2, we have been allotted with two hour long sessions per crew and week, that is, the entire crew will participate in the experiment at the same time. This adds up to 6 sessions in total for the whole phase. Similarly to phase 1, one of the sessions will be performed with the assistance of the VA, and the other without.

The exact design for each experiment session is yet to be determined. For example, we still have to decide how many anomalies are we going to introduce in each scenario. Although we initially planned to go for five, the pilot experiments at our lab have shown that this might be too much, as explained further in this document. However, if anything, the design is going to be very similar to the one for the pilot experiments (except for the fact that they will take place at the actual HERA analog). To avoid repetition, only the latter is explained in the next section.

4.2. Pilot experiments

We aim to conduct several series of pilots at our lab. The main goals of such experiments are:

- *Testing the software*: We want to ensure that our software is user resistant and bug free.
- *Testing the experiment set up*: We want to ensure that the experiment set up design is consistent and well scaled to the available resources (time and material wise).
- *Testing the experiment procedures*: We want to check that the procedures for the experiments are clear and coherent with the experiment set up..
- *Improving the VA*: We aim to retrieve some knowledge about which are the key issues to tackle in order to improve the VA.

Unfortunately, the experiments at the lab are doomed to lack the fidelity of the environment that the HERA analog provides. Additionally, we will be hiring subjects that will not be a trained crew, and they will evidently not be in the psychological conditions that a prolonged status of confinement and isolation produces. Hence, the pilot experiment conditions will not be as faithful to the HERA ones as we would like, but this should not be an obstacle to pursue any of the goals above stated.

In fact, we have already performed a first series of pilot experiments. We recruited a total of six subjects and used our lab facilities to mimic the HERA experiments set up to some extent. Such set up was quite simple:

1. The subject sat on a desk with two computers: one to display and interact with NASA's ECLSS screen, and the other with the Daphne-AT VA on it.
2. The subject was provided with a thorough tutorial to:
   (a) Briefly present the aim and set up of the experiment.
   (b) Explain what are the tasks that he/she had to perform.
   (c) Train him/her to use the ECLSS screen.
   (d) Train him/her to use Daphne-AT.
3. The subject was exposed to a set of five anomalies that were previously designed. Such anomalies were simulated with NASA's sECLSS software.
4. For each anomaly, the subject had to attempt to solve it using all the available resources to him/her.

Since we did not have the physical embodiment for the HERA analog, we created a task to simulate the process of resolving an anomaly. Each anomaly can be treated by applying a certain set of procedures -the ones that NASA provided us with, supported in PDF format-. For each procedure, we broke it down to a series of steps, wrote each step on a card, and by grouping all the step cards, we obtained a card deck for each procedure.

The card decks for all the procedures were made available to the subject at the beginning of each experiment (they were located in the same desk than the computers). Each card deck was shuffled beforehand -that is, the steps were not in the correct order-. The task of the subject consisted of:

1. Paying close attention to the ECLSS screen and Daphne-AT to detect new anomalies.
2. Using the ECLSS screen and Daphne-AT to identify the anomaly that is happening.
3. Using Daphne-AT to decide which set of procedures should be applied.
4. Finding all the (shuffled) card decks that correspond to each of the selected procedures.
5. Put the step cards in each deck in the correct order.
6. Communicate the selected procedures and the last step of each procedure to the supervisor of the experiment.

An anomaly was terminated from the sECLSS simulator if the subject correctly solved it (which meant both selecting the correct procedures to be applied and providing the correct last step for each of them) or if a certain time limit was reached (chosen so that the overall duration of the experiment did not exceed one hour). The experiment finished when the last of the five anomalies was terminated.

The card deck system allowed us to:

1. Simulate the mental workload that applying a procedure with the real HERA equipment would have.
2. Simulate the estimated time of resolution that treating each anomaly requires.
3. Have a success/fail criteria to decide whether the subject has successfuly resolved an anomaly or not.

At the end of the experiment, the subjects were requested to answer a set of standard surveys in order to subjectively assess their situational awareness (SART [4] survey), their trust in the VA (JIAN's [5] survey) and the perceived workload (NASA's TLX [6] survey).

(*Note*: By the time of the writing of this document, we have only conducted sessions in which the subject was equipped with the VA. We should also perform experiments without it in order to compare both the subject performance during the anomaly treatment process and the results from the surveys. The only reason we did not do that is because we had a tight project schedule, and an imminent software and documentation delivery for NASA. We were more concerned about testing the software and the experiment set up before the delivery than about retrieving science from the pilot experiments. We have already done such delivery, and we still have plenty of time until the HERA experiments to this.)

4.3. Pilot experiments outcome and results

As expected, the pilot experiments were extremely useful to detect bugs in the software. No major issues appeared regarding this topic.

The answers from all the surveys can be found in the Appendix of this document. The remarks to be made after analyzing the results for each one of them are as follow:

- *From the SART survey (about the situational awareness)*

  The subjects rated the anomaly scenarios as complex and changeable. They stated that they required them to stay very alert, and that they struggled to focus on more than one task at the time.

- *From the JIAN's survey (about the trust in the VA)*

  Overall, the subjects trusted Daphne-AT a lot. Despite that, Daphne-AT did not get very good marks for its reliability. Also, all subjects stated that they were very unfamiliar with Daphne-AT.

- *From the NASA's TLX survey (about the workload)*

  In general, the subjects indicated that the workload was relatively high, although the variance on their answers is considerable. Also, there was a consistent agreement between subjects on the fact that the task was temporally demanding.

Regarding the experiment design, from the survey analysis above presented and the feedback that we got from the subjects, the main outcome was that all of them were overwhelmed by the overall situation. Our main concerns in terms of what we should improve for future experiments in the lab (and perhaps at HERA too) are:

- The subjects struggled in understanding how to use the VA properly.
- The subjects had a rough time in applying a procedure fast enough.
- The subjects had problems with paying attention to both screens (the ECLSS one and the VA one) while applying a procedure.

The first issue is certainly a non neglectable obstacle. Perhaps we will have to extend the duration of the sessions and put more emphasis on the tutorial stage. However, we are not very concerned about this issue for the HERA experiments, since we will have specific sessions (adding up to a total of eight hours) which only purpose will be to train the crew to use the VA. Additionally, we will be dealing with a trained crew. We are confident these facts will suffice to avoid this problem.

To solve the third issue, we added the alarm sounds to the VA (they were not implemented before the pilot experiments). This way, the user can focus on the anomaly response task, and trust Daphne-AT to bring his/her attention back to the screen via a sound whenever a new anomalous information appears on the screen.

Finally, regarding the second issue, we initially thought that the “deck card system” was too straightforward, and that the subjects would perform the task too fast and with little effort. We were also convinced (and still are) that this task was far more simple than performing the actual procedure at the HERA analog, and hence, if anything, any anomaly would take longer to be resolved at HERA than at our lab. All these considerations make of this issue perhaps the most relevant one. To tackle it, we are considering reducing the number of anomalies that each subject will be exposed to during a session. We have not made a final decision yet though.

5. Future work

This project is meant to be four years long, and we are barely in the seventh month. Hence, it is still at an early phase, and there is plenty of room for improvement. In this section, the main topics that should be tackled are discussed.

5.1. Improving the GUI

The GUI shown in Section 2.1 is the result of the very first iteration. We still have to perform more pilot experiments to get feedback from the subjects. In order to improve and maximize the enhancement in human performance that the VA is supposed to provide, and according to the original proposal, the following topics must be addressed:

- *What information should be displayed on the GUI?*
  (e. g.: Should the VA provide explanations for its decisions? Would that improve the user's situational awareness? Would it be too much information on the screen?)

- *How should information be displayed in the GUI?*
  (e. g.: Should the windows from the display be minimizable? Should they be displayed in a different order? Is the color code intuitive?)

- *Automated vs. mixed collaboration approach.*
  (e. g.: Should the VA take more initiative? Would that improve human performance? Would that decrease the user's trust in the VA?)

- *User customization.*
  (e. g.: Should we allow the user to accommodate all the previous items according to its preferences?)

5.2. Creating a virtual reality tool

There is probably no better environment than the HERA analog to test the VA. However, such facilities will be rarely available to us. For this reason, we need to develop some sort of tool (a better approach than the “card deck system” one) to mimic the experiment conditions at the analog at our lab and perform more tests in the future, specially after the HERA campaign.

To do so, from the very beginning of the project we started to develop a Virtual Reality (VR) tool that would allow the user to be immersed in a virtual model of the HERA analog. A model for its exterior (depicted in Figure 16) is publicly available on the web, but its interior is still empty and to be developed.

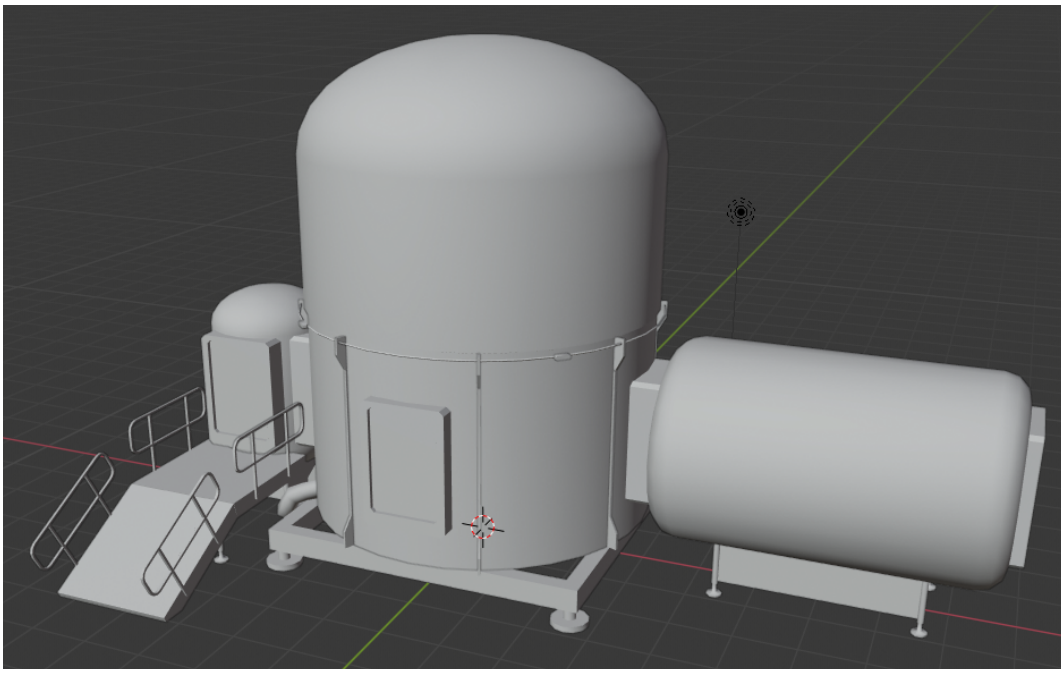

Figure 16: The VR model for the exterior of the HERA analog (publicly available).

Our current approach is to build a set of blocks inside the HERA model in order to mimic the actual ECLSS subsystems. That is, when using the VR tool, the user could “walk” around the interior of the HERA analog model and find a square block wherever a particular subsystem should be located. Such blocks would support some sort of interactive feature with the purpose of simulating the anomaly resolution process. In the current version of the VR tool, such interactive feature consists of a simple navigator of the same step break down that we used in the pilot experiments. In the future, we plan on reproducing all the actions that the actual procedures involve (replacing a component, opening a box, turning down switches, etc.).

5.3. Developing a new simulator

As explained in Section 3.1, the current simulator is not model based, and hence, it has a very low fidelity. Because the simulated data is unrealistic, the results obtained using the ECLSS cannot used to assess and validate the performance of the detection and diagnosis skills of the VA under real conditions. Additionally, being the data so simple, building sophisticated detection and diagnosis skills might be a (perhaps futile) overshot.

We plan on building a new simulator of our own to be used in the experiments at our lab (not at HERA, since we are compelled to use the ECLSS one there). Such simulator would be model based, and it would allow us to:

- *Validate the VA performance in a real scenario.*
- *Improve the detection and diagnosis skills.*
- *Build an historical database.*
- *Emulate a real time interaction between user and ECLSS environment.*

Regarding the last item, remember that NASA's sECLSS simulator requires an external agent to manually phase in and phase out the anomalies, and that the interaction between subject and simulator is unidirectional (the simulator outputs the simulated sensor data, but the actions made by the Daphne-AT user do not affect at all the status of the simulation).

If this simulator was integrated both with Daphne-AT and the VR tool, this could provide an interesting and complete framework to test the VA and simulate the HERA ECLSS environment. For example, an anomalous scenario could be completely defined via a configuration file, in which the anomalies that have to be simulated are specified. Then the simulation would be triggered, and whenever the user encountered an anomaly, it would start interacting with the HERA subsystems via the VR tool. During the anomaly resolution process, the status of the simulation would change depending on how well the user performs a determined procedure, gradually improving the sensor data readings until they go back to nominal, or worsening them in case the user does something wrong.

5.4. Improving the detection and diagnosis skills

The current detection skill is limited to a basic threshold check. Despite being quite elementary, it is very robust and reliable, and its more than sufficient given the current simplicity of the simulated sensor data. That being said, if a better and more realistic simulator was to be built, the detection skill could be improved.

For instance, we have already implemented more sophisticated techniques for anomaly detection in another project of the SEAK lab [2], as mentioned in the Preliminaries of this thesis. These approaches include, among other, Auto Regressive Integrated Moving Average (ARIMA) and Adaptive Kernel Density-based methods. However, such implementations were focused on static anomaly detection analysis (not in real time), and we still have to figure out how to integrate them to the Daphne-AT tool so that they are computionally feasible.

Ultimately, no single algorithm works best for all the different possible data sets. Therefore, we plan on implementing more approaches and use a conservative OR rule, flagging an event as anomalous if any of the multiple criteria determines it as so.

Regarding the diagnosis skill, the current approach has proved to work fairly well. However, if an anomaly historical database was to be implemented in the future, it would be interesting to study a data driven approach in addition to the current rule based one. This would allow Daphne-AT to better infer what root cause might be producing an anomaly by comparing the current signature to past ones. In addition to that, a data driven diagnosis skill would make Daphne-AT more versatile too: while the rule based strategy requires to characterize and hard code all the set of rules, the data approach only requires a proper labeled data set, and hence, Daphne-AT could be easily adapted to other anomaly treatment processes not necessarily related to the ECLSS system.

6. Conclusions

We have presented why and how crew and spacecraft autonomy are going to be a critical obstacle towards mission success in future deep space human exploration missions. To tackle such issue, and according to NASA's interests, we have proposed to develop a VA to assist the crew in the context of anomaly treatment without mission control ground support. In particular, we have focused in anomalies related to the ECLSS system of a spacecraft.

In this document, we have thoroughly described the implementation of the said tool, named Daphne-AT. First, we have presented its GUI in detail, and after that, we have not only explained the software and resources that support it, but also the motivations that ultimately lead to the current version of the VA. Additionally, we have designed and presented a set of experiments that have served and will serve to test the VA in the close future.

Unfortunately, and because of the timeline of the project and this thesis itself, few results have been presented regarding the usefulness of the VA. We have conducted some pilot experiments to test ifs first iteration though. The outcome of such experiments was rather satisfactory, since the software proved to be robust enough and we got useful feedback from the subjects. However, we still have to perform more pilot experiments without the VA, in order to assess if it indeed enhances human performance or not.

Overall, we have successfully developed a working baseline version of the VA. Also, we have identified the key features that require further work, and we have set what we believe is a promising path towards a complete tool for anomaly detection in deep space human exploration missions.

References

[1] Virós Martin, A. (2017). *Design and Development of a Cognitive Assistant for the Architecting of Earth Observing Satellites* (Final Bachelor's Thesis), Universitat Poliltècnica de Catalunya (Barcelona, Spain) & Cornell University (Ithaca, New York, USA). https://upcommons.upc.edu/bitstream/handle/2117/108999/126671.pdf?sequence=1&isAllowed=y .

[2] García Ling, C. (2018). *Daphne: A Tool for Anomaly Detection* (Final Bachelor's Thesis), Universitat Poliltècnica de Catalunya (Barcelona, Spain) & Cornell University (Ithaca, New York, USA). https://upcommons.upc.edu/bitstream/handle/2117/121052/memoria.pdf?sequence=1&isAllowed=y .

[3] Dutta, P., Balcells-Quintana, O., Virós Martin, A., Whittle, R., Josan, P. K., Beebe, N., Dunbar, B. J., K. W. Wong, R., Diaz-Artiles, A., Selva, D. (2020). *Virtual Assistant for Anomaly Treatment in Long Duration Exploration Missions.* In AIAA SciTech Forum. https://doi.org/doi:10.2514/6.2020-2255 .

[4] Endsley, M. R., *Measurement of Situation Awareness in Dynamic Systems*, Vol. 37, No. 1, 1995, pp. 65–84.

[5] Hart, S. G., and Staveland, L. E., *Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research*, Human Mental Workload, Vol. 52, edited by P. A. Hancock and N. B. T. A. i. P. Meshkati, North-Holland, 1988, pp. 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9 , http://www.sciencedirect.com/science/article/pii/S0166411508623869 .

[6] Jian, J.-Y., Bisantz, A. M., and Drury, C. G., *Foundations for an Empirically Determined Scale of Trust in Automated Systems*, International Journal of Cognitive Ergonomics, Vol. 4, No. 1, 2000, pp. 53–71. 10.1207/S15327566IJCE0401_04,URLhttps://doi.org/10.1207/S15327566IJCE0401{_}04 .

[7] Virós Martin, A., Selva, D. (2020). *Explanation Approaches for the Daphne Virtual Assistant.* In AIAA SciTech Forum. https://arc.aiaa.org/doi/pdf/10.2514/6.2020-2254 .

[8] Virós Martin, A., Selva, D. (2019). *Daphne: A Virtual Assistant for Designing Earth Observation Distributed Spacecraft Missions.* In IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. https://ieeexplore.ieee.org/document/8901436 .

[9] Virós Martin, A., Selva, D. (2019). *From Design Assistants to Design Peers: Turning Daphne into an AI Companion for Mission Designers*. In AIAA SciTech Forum. `https://arc.aiaa.org/doi/10.2514/6.2019-0402` .

[10] Bang, H., Virós Martin, A., Selva, D. (2018). *Daphne: An Intelligent Assistant for Architecting Earth Observing Satellite Systems*. In AIAA SciTech Forum. `https://arc.aiaa.org/doi/10.2514/6.2018-1366` .

[11] Crawley, E., Cameron, B., and Selva, D., *System Architecture: Strategy and Product Development for Complex Systems*, Prentice Hall, 2015. `https://books.google.ru/books?id=67TuoQEACAAJ` .

[12] Honnibal, M., Johnson, M., *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2015. `https://www.aclweb.org/anthology/volumes/D15-1/` .

[13] François, C., *Keras*, Github, 2015. `https://github.com/keras-team/keras` .

[14] Engelbart, D. C., *Augmenting human intellect: a conceptual framework*, Tech. rep., Stanford Research Institute, WashingtonD.C., 1962.

[15] Onken, R., and Walsdorf, A., *Assistant systems for aircraft guidance: Cognitive man-machine cooperation*, Aerospace Science and Technology, Vol. 5, No. 8, 2001, pp. 511–520. doi:10.1016/S1270-9638(01)01137-3

[16] Donath, D., Rauschert, A., and Schulte, A., *Cognitive assistant system concept for multi-UAV guidance using human operator behaviour models*, Humous'10, 2010. `http://www.dagstuhl.de/mat/Files/11/11141/11141.StrenzkeRuben.Other1.pdf`

[17] Tokadli, G., *Cognitive Assistant — ACSL*, , 2017. `http://www.imse.iastate.edu/acsl/cognitive-assistant/`

[18] Tokadli, G., and Dorneich, M. C., *Development of Design Requirements for a Cognitive Assistant in Space Missions Beyond Low Earth Orbit*, Journal of Cognitive Engineering and Decision Making, 2017, p. 1555343417733159

[19] Tokadli, G., and Dorneich, M. C., *Development of a Functionality Matrix for a Cognitive Assistant on Long Distance Space Missions*, Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Vol. 61, SAGE Publications Sage CA:Los Angeles, CA, 2017, pp. 247–251

[20] Tokadli, G., and Dorneich, M. C., *Envisioned Concept of Operations for Beyond Low-Earth Orbit: The Collaborative Decision Making Between Human and Cognitive Assistant*, AIAA SPACE and Astronautics Forum and Exposition, 2017, p. 5270

Appendix

SART survey

Table 1 entails the results from the SART survey. This survey involves a self rating technique which elicits the subjective opinion on how aware a subject was during the performance of a certain task. The subject is asked provide a grade (ranging from 1 to 7) to a series of statements.

Table 1: Results from the SART survey. The grading scale ranges from 1 to 7. A total amount of six subjects were surveyed.

Question	**Average**	**STD**
How changeable are the anomaly scenarios? Are the anomaly scenarios highly unstable and likely to change suddenly (High) or are these very stable and straightforward (Low)?	4,00	1,26
How complicated are the anomaly scenarios? Are these complex with interrelated components (High) or are these simple and straightforward (Low)?	4,33	2,16
How many variables are changing within the anomaly scenarios? Are there a large number of factors varying (High) or a very few variables changing (Low)?	4,00	1,10
How aroused are you during the anomaly treatment? Are you alert and ready for activity (High) or do you have a low degree of alertness (Low)?	4,67	1,86
How much are you concentrating on the anomaly treatment? Are you concentrating on many aspects of the anomalies (High) or focused on only one (Low)?	3,83	2,32
How much is your attention divided during the anomaly treatment? Are you concentrating on the many aspects of the anomaly scenarios (High) or focused on only one (Low)?	1,50	1,22
How much mental capacity do you have to spare during the anomaly treatment? Do you have sufficient to attend to many variables (High) or nothing to spare at all (Low)?	3,67	1,63
How much information have you gained about the anomaly scenarios? Have you received and understood a great deal of knowledge (High) or very little (Low)?	3,67	1,75
How familiar are you with the anomaly scenarios? Do you have a great deal of relevant experience (High) or is it a new situation (Low)?	1,17	0,41

JIAN's survey

Table 2 entails the results from the JIAN's survey. This survey evaluates the trust of the subject while interacting with an automated system. The subject is asked provide a grade (ranging from 1 to 7) to a series of statements.

Table 2: Results from the JIAN's survey. The grading scale ranges from 1 to 7. A total amount of six subjects were surveyed.

Question	**Average**	**STD**
Daphne-AT is deceptive.	2,17	1,17
Daphne-AT behaves in an underhanded manner.	2,17	1,47
I am suspicious of Daphne-AT's intent, action or outputs.	1,67	1,63
I am wary of Daphne-AT.	2,00	0,63
Daphne-AT's actions will have a harmful or injurious outcome.	1,50	0,55
I am confident in Daphne-AT.	5,33	1,21
Daphne-AT provides security.	3,83	1,72
Daphne-AT has integrity.	5,33	1,21
Daphne-AT is dependable.	4,33	1,51
Daphne-AT is reliable.	4,00	1,26
I can trust Daphne-AT.	4,83	1,17
I am familiar with Daphne-AT.	1,83	0,98

NASA's TLX survey

Tables 3 and 4 entail the results from the NASA's TLX survey. This consists of a technique to assess the relative importance of six factors that determine how much workload did the subject experience while performing a certain task. Such factors are:

- *Mental demand*: How much mental and perceptual activity was required (e.g. thinking, deciding, calculating, remembering, looking, searching etc.)? Was the task easy or demanding, slow or complex, exacting or forgiving?

- *Physical demand*: How much physical activity was required (e.g. pushing, pulling, turning, controlling, activating, etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?

- *Temporal demand*: How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?

- *Performance*: How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?

- *Effort*: How hard did you have to work (mentally and physically) to accomplish your level of performance?

- *Frustration level*: How insecure, discouraged, irritated, stressed, and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

First, the subject is asked to rank the importance of each factor by itself on a 0 to 100 scale. Then, for each combination of two factors, the subject is asked to evaluate which one was more relevant while performing the task.

Table 3: Results from the NASA's TLX survey (first part, *grading*). The grading scale ranges from 0 to 100. A total amount of six subjects were surveyed.

Question	**Average**	**STD**
Mental demand	71,33	19,42
Physical demand	21,67	17,28
Temporal demand	73,00	23,32
Performance	31,00	21,59
Effort	68,17	19,38
Frustration level	49,67	28,57

Table 4: Results from the NASA's TLX survey (second part, *comparing*). The answers can be either 1 (first option) or 2 (second option), depending on which one the user regards as more relevant. A total amount of six subjects were surveyed.

Question	**Average**	**STD**
Effort *vs.* Performance	1,67	0,52
Temporal demand *vs.* Frustration level	1,17	0,41
Temporal demand *vs.* Effort	1,17	0,41
Physical demand *vs.* Frustration level	1,83	0,41
Performance *vs.* Frustration level	1,67	0,52
Physical demand *vs.* Temporal demand	2,00	0,00
Physical demand *vs.* Performance	1,67	0,52
Temporal demand *vs.* Mental demand	1,50	0,55
Frustration level *vs.* Effort	1,67	0,52
Performance *vs.* Mental demand	1,67	0,52
Performance *vs.* Temporal demand	1,50	0,55
Mental demand *vs.* Effort	1,00	0,00
Mental demand *vs.* Physical demand	1,17	0,41
Effort *vs.* Physical demand	1,00	0,00
Frustration level *vs.* Mental demand	2,00	0,00

Note from the author

Now that the Daphne-AT tool has been presented in detail, I want to make some disclaimers regarding the content in this thesis.

I want to remark that this was a collaborative project and by no means I claim the authority of all the software and resources presented in this document.

In particular, as already mentioned in the Preliminaries section, most of the software infrastructure that supports Daphne-AT was retrieved from Daphne, a tool developed by Antoni Virós. I simply took such tool, removed all the features and functionalities that were intrinsic to the constellation design problem, and then added the ones corresponding to the anomaly treatment one.

Furthermore, although I strongly contributed to the development of the neo4j database in many ways, I want to acknowledge that the leads of such task were other lab teammates, namely Nikita Beebe and Kyle York.